Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

183

The Munich Project CIP

Volume I: The Wide Spectrum Language CIP-L

By the CIP Language Group:
F. L. Bauer, R. Berghammer, M. Broy, W. Dosch,
F. Geiselbrechtinger, R. Gnatz, E. Hangel, W. Hesse,
B. Krieg-Brückner, A. Laut, T. Matzner, B. Möller, F. Nickl,
H. Partsch, P. Pepper, K. Samelson (†), M. Wirsing
and H. Wössner

Springer-Verlag
Berlin Heidelberg New York Tokyo

Authors

F. L. Bauer
R. Berghammer
W. Dosch
R. Gnatz
E. Hangel
B. Möller
H. Partsch
P. Pepper
K. Samelson (†)
H. Wössner
Institut für Informatik der TU München
Postfach 20 24 20, 8000 München 2

M. Broy
F. Nickl
M. Wirsing
Fakultät für Mathematik und Informatik, Universität Passau
Postfach 2540, D-8390 Passau

F. Geiselbrechtinger
Department of Computer Science, University College Dublin
Belfield, Dublin 4, Ireland

W. Hesse
Softlab GmbH
Arabellastr. 13, D-8000 München 81

B. Krieg-Brückner
Fachbereich 3, Informatik, Universität Bremen
Postfach 33 04 40, D-2800 Bremen 33

A. Laut
PCS GmbH, Periphere Computer-Systeme
Pfälzer-Wald-Str. 36, D-8000 München 90

T. Matzner
sd&m GmbH, Software Design & Management
Führichstr. 70, D-8000 München 80

CR Subject Classification (1982): D.1.0, D.2.1, D.2.4, D.3.1, D.3.3, F.3.1, F.3.2, F.3.3

ISBN 3-540-15187-7 Springer-Verlag Berlin Heidelberg New York Tokyo
ISBN 0-387-15187-7 Springer-Verlag New York Heidelberg Berlin Tokyo

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.
2145/3140-543210

PREFACE

This book is the first of two volumes that present the main results having emerged from the project CIP - Computer-Aided, Intuition-Guided Programming - at the Technical University of Munich. The central theme of this project is program development by transformation, a methodology which is felt to become more and more important.

Whereas Volume II will contain the description, formal specification, and transformational development of a system, CIP-S, that is to assist a programmer in this methodology, the present volume gives the description and formal definition of a program development language CIP-L designed particularly for use in transformational development of programs from formal specifications. Many sources have influenced the development of this language over the past eight years, and we feel that it has now matured and consolidated to a degree that its study will be profitable to others.

Three aspects of this language appear to be of special interest:

First, the language is a *coherent wide spectrum language*. This means that it comprises a number of expressive levels ranging from predicative and algebraic specifications over applicative and procedural constructs (including parallelism) down to a machine-oriented level using jumps and pointers. However, these levels do not just form a loose collection of features; rather they are closely linked by formal transformation rules that relate the various constructs to give a coherent semantics to the entire language. Besides, these rules also give a guideline for the overall structure of the transformational development process. Since CIP-L comprises most of the essential concepts of today's programming languages in more or less similar form, the rules also provide some insight into the general structure of programming languages and programming.

The second major aspect of interest in this book is the definition of the language by the new method of *transformational semantics*. In this approach, a kernel language is distinguished as that part of the language which contains the essential semantic concepts. All other language constructs are then mapped into this kernel by formal transformation rules that allow reducing every program to an equivalent one of the kernel language. It seems remarkable that the set of rules needed for the particular language CIP-L is quite small and easy to survey; in most cases just one rule per additional language construct suffices. Thus, a transformational definition of a well-designed language is as concise as, say, a denotational one; moreover, in our opinion, it is much easier to comprehend.

As a third major aspect, CIP-L is an *abstract scheme language*. This means, first, that its formal definition works on the abstract syntax of the language. As a consequence, CIP-L allows various concrete syntactic representations, two of which, viz. an ALGOL-like and a PASCAL-like form, are given in syntax charts in an appendix. Second, CIP-L is a scheme language in that it is independent of any particular set of data structures (except the truth values); rather it comprises elaborate means for defining new data structures.

The book is organized as follows: Part 0 contains a general introduction to transformational programming and to the project CIP in particular. In Part I, the various constructs of the language are introduced informally together with examples of their place in program development. Part II, the heart of the book, then gives a description of the language in a systematic way. However, readability is considered more important than complete formality; in particular, the transformation rules are given in concrete syntax, and a number of self-evident context conditions are omitted. Finally, Part III contains a complete formal definition of the language in the same order of presentation as Part II.
Cross-references within one part are given by section numbers only; references to other parts are made by prefixing the respective section numbers with the (roman) part numbers.

We would like to express our thanks to the Deutsche Forschungsgemeinschaft who has sponsored this research within the Sonderforschungsbereich 49 "Programmiertechnik" during the past nine years. Also, we gratefully acknowledge valuable criticism of the language and of its definition by the lecturers and participants of the Marktoberdorf Summer Schools 1978 and 1981, notably by E.W. Dijkstra, D. Gries, and C.A.R. Hoare, as well as by the members of IFIP Working Group 2.1, notably by H.J. Boom, G. Goos, L.G.L.T. Meertens, S.A. Schuman, and M. Sintzoff. Particular thanks are due to R.S. Bird for pointing out a severe error in an earlier version of the semantic specification of the kernel language. Last, but by no means least, we gratefully acknowledge many helpful remarks by our colleagues C. Delgado Kloos, H. Ehler, F. Erhard, U. Hill-Samelson, A. Horsch, H. Hußmann, W. Meixner, R. Obermeier, H.-O. Riethmayer, G. Schmidt, and R. Steinbrüggen.

Munich, December 1984

The CIP Language Group

TABLE OF CONTENTS